Estruturas de Dados E Algoritmos I

Trabalho 2

Docente: Gil Catarino nº32378 **Discente: Lígia Ferreira**

Introdução

O Objectivo deste trabalho é de desenvolver um corrector ortográfico, através da linguagem Java e dos tipos abstratos de dados leccionados implementar tabelas de dispersão utilizando a implementação de HashTable, LinearHashTable, QuadraticHashTable, DoubleHashTable.

Resolução do Problema

Depois de ter as devidas implementações feitas das HashTables, começou-se por criar o método ***LerFicheiro(String file1)*** que lê um ficheiro dado como "input" pelo user, que neste caso é a lista de palavras, e de colocar todas essas mesmas palavras numa QuadraticHashTable, sendo este o nosso "dicionário" de palavras ("dicFile.txt).

De seguida criou-se o método ***void LerDoc (String file3)*** , este que lê também uma linha com texto fornecido pelo o utilizador ("docFile.txt"), faz split() para retirar os espaços entre as palavras na frase, e guarda todas essas palavras num ***ArrayList<String> array1*** .

Entretanto foi criado um método que inicialmente tinha como nome MostraCertos(String docFile), mas que mais tarde seria trocado por ***void spellCheck(String docFile)*** , este que vai receber esse ficheiro texto guardado no array1, e verificar primeiro quais dessas palavras estão e não estão no dicionário de palavras. As que não estão são dadas como "mal-escritas" e então vai verificar se já estão na tabela ***HashAberto hashSuj*** que é uma hash que usa os mesmos métodos que uma "hash" normal, mas que foi implementada para suportar ArrayList, e trabalhar com ArrayList em vez dos "elementos" criados para ser usados nas "hash normais". Depois, de verificar se está ou não nessa hashSuj , se estiver, passa por diferentes métodos (***static ArrayList<String> Remove1Letra(String s) ; static ArrayList<String> AlteraAdjacentes(String s); static ArrayList<String> Adciona1Letra(String s)***) , que por sua vez são passadas as Strings dadas como incorretas, e que como o próprio nome indicam, percorrem a string, e removem/adicionam ou trocam de adjacentes por cada char encontrado. Isto de maneira a alterar a string de todas as maneiras possíveis, e guardando previamente em arrayList criadas nesses métodos.

Quase por terminado, é criado um ***ArrayList<String> sugestão***, que cada vez que encontrar uma String anteriormente alterada e guardada nos arrays nos métodos anteriores descritos, e encontrar no dicionário dado inicialmente (dicFile), é colado nesse array (sugestão).

Por fim, é criado um ficheiro "errorFile.txt", onde por cada palavra errada, vai printar a palavra errada -→ e a sugestão obtida , uma por linha, de todas as palavras dadas inicialmente como incorrectas. Desta forma, o user consegue aderir ao ficheiro na pasta, e verificar todas as palavras que se enganou a escrever, e ver como poderia acertar ☺